



TITLE:

複数マイクロプロセッサによる待 行列網シミュレータ:HASS-QN (待 行列理論とその応用)

AUTHOR(S):

宮原, 秀夫

CITATION:

宮原, 秀夫. 複数マイクロプロセッサによる待行列網シミュレータ:
HASS-QN (待行列理論とその応用). 数理解析研究所講究録 1981, 425:
92-105

ISSUE DATE:

1981-04

URL:

<http://hdl.handle.net/2433/102605>

RIGHT:

複数マイクロプロセッサによる待行列網シミュレータ (HASS-QN)

阪大 基礎工 宮原 秀夫

1. まえがき

近年、コンピュータネットワークや複合計算機システムなど、複数のリソースを有機的に結合して構築されるシステムを解析評価する手段として、待行列網理論(QNTh.)がその成果を上げてきているが、まだ多くの面で適用範囲が限られており、シミュレーションに頼ることが多い。

本報告では、待行列網(QN)にモデル化できるシステムを解析するための、マイクロプロセッサを複数台使用した専用シミュレータの構成を中心に述べる。

多重プログラミング環境における計算機システムに対するスループットの評価、ターンアラウンドタイムの推定、あるいはパケット交換方式におけるコンピュータネットワークでのメッセージ伝送遅延の推定、ルーティング方式の評価、衛星通信チャネルへのアクセス方式の評価などにおいて、待行列理論が盛んに用いられている。しかし QNTh. を用いて陽な

解析解、あるいは実用的な解が得られるモデルは限られている。その中で特に、ブロッキングのある制限型モデル、さらには、ブロッキングの伝播によってシステムデッドロックが生ずる様なモデルに対しては、特性の解析、評価は現在のところ不可能である。従ってこの様なモデルに対しては、汎用計算機を用い、適当なシミュレーション言語 (GPSS, SOL, SIMSCRIPT) でシミュレーションを行う方法がよくとられる。

汎用言語によるシミュレーションには次のような問題点があるように思われる。

- (1) ネットワークタイプのモデルに対しては、そのプログラミングにかなりの精熟度と時間を要し、シミュレーションの妥当性を十分検討する時間が少なくなる。
- (2) ほとんどのプログラムがバッチ的に処理されるため、シミュレーション途中で、システムのパラメータを変更したり、ある時点でのシステム状態を出力したりする、いわゆるマン・マシン・インタラクティブなシミュレーションを実現できない。
- (3) 多くの興味あるシステムは、潜在的に並列性を含んでいるが、シミュレーションのモデル化の際、それを反映しにくい。
- (4) モデルに対する多少の変更、例えば、あるノードでの処

理規律の変更、ブランチの追加あるいは削除、に伴うプログラムの変更に時間を要し容易に行えない。

そこで以上のような問題を考えて、本来シミュレーションの対象となるシステムがもっている並列性を有効に利用できるように、低価格、低速のマイクロプロセッサを複数台結合して、並列プロセスを各プロセッサに写像できるようにし、会話型式でシミュレーションを実行できる機能を有するQN専用シミュレータ HASS-QN (HAndai System Simulator for Queueing Network)の開発を行った。

2. HASS-QNの機能

今回試作中の HASS-QN は、time driven方式を採用し、対象とし得るモデルは表1に示す通りである。

表1. 現在の HASS-QNの機能

対象とするモデル	複数ノードから成るネットワーク行列 (Open, closed)
ノード	Single serverと、行列 (最大長可変)
サービス規律	同一優先順位内で FCFS
客のタイプ	優先順位をもつ客
サービス時間	指数、アーラン、超指数分布、単位分布
系外からの到着	ポアソン到着、一定到着

なお今後、上記以外のより一般的モデルに対して適用できるように、システムを拡充していく予定である。

3. HASS-QNのハードウェア構成

3.1 プロセッサ機能

今回試作中のHASS-QNは、13組のシングルボードマイコン(8ビット系-12組, 16ビット系-1組)と、8ビット系のマイコン2台、共有メモリ、ボード1枚、バスコントロールボード1枚より成る。その全体の構成は図1の通りである。

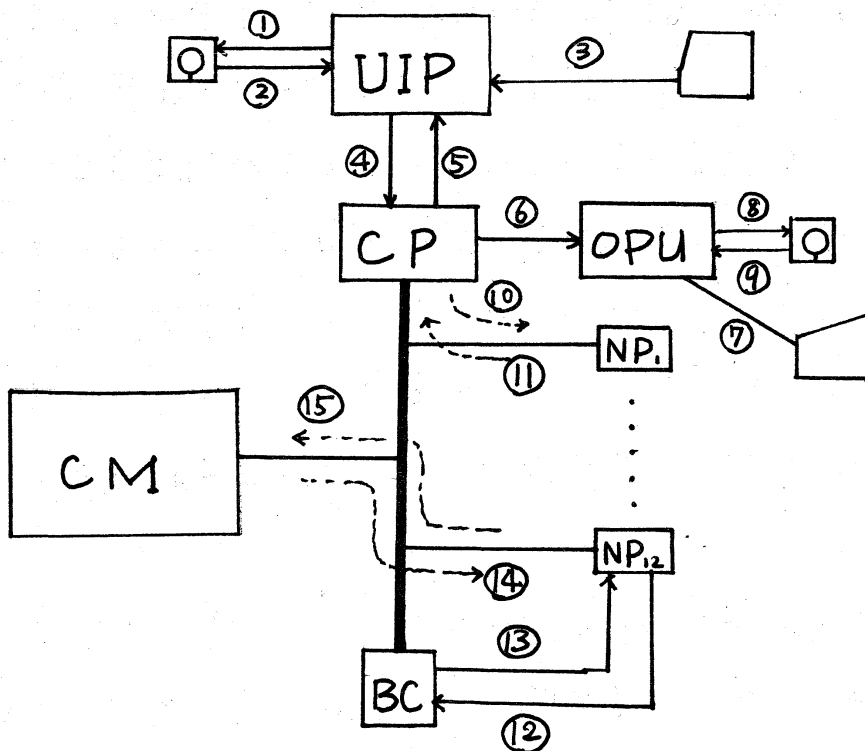


図1 HASS-QNのシステム構成

- (1) NP (Node Processor) 8ビット系マイクロプロセッサ
Z-80使用のシングルボード12組。QNにおけるサーバーとQueueに対応。
- (2) CP (Control Processor) 16ビット系マイクロプロセッサ
Z-8000使用のシングルボード1組。シミュレーション全体の管理を行う。具体的には、シミュレーションの起動、NPへのプログラムの転送、時刻管理、トランザクションの発生、各ノードからの統計量を吸い上げ統計処理用のプロセッサへの転送などを行う。
- (3) UIP (User Interface Processor) 8ビット系マイコン：
CPU 8080 : 2次記憶 : フロッピーディスク : I/O装置
ユーザが会話的なシミュレーションが行えるようにするためのプロセッサで、NPプログラムへのパラメータ入力、2次記憶からのCPへのプログラム転送、CPの制御、NPプログラムの開発、シミュレーションの中断、再開、システムパラメータの変更指令などを行う。
- (4) OPU (Output Processing Unit) 8ビット系マイコン、
CPU: Z-80 : 2次記憶 : フロッピーディスク : I/O。
リアルタイム (シミュレーション時における) な統計処理をする。試行錯誤的なシミュレーションを可能に

すなわち、シミュレーション実行中のシステムの状態をCRTへ表示する。

- (5) CM (Common Memory) 64 Kbits の Dynamic RAM
各NP, CP 間の通信はすべてこの CM を通して行われ
る。その方式については後述する。
- (6) BC (Bus Controller) 各NP, CP からのバス競合を防
ぐためバスの管理、メモリアクセスのスケジューリン
グを行う。

各エント間のつながりを図1の中の番号に対応させて説明する。

- ① 各NPのプログラム(ユーザがUIP上で開発した)の転送
- ② 各NPのプログラム(今からシミュレーションを行うためCP, NP_sに送る)の転送
- ③ システムパラメータの入力
- ④ NP_s, CPのプログラム転送, シミュレーションの中断(割り込み)
- ⑤ シミュレーション中断, 終了時の主導権の委譲
- ⑥ NPから集収した統計量の転送
- ⑦ シミュレータから送られた統計量でOPUで処理された情報の表示

- ⑧ シミュレータからの統計量の格納
- ⑨ 統計計算のための統計量の入力
- ⑩ NP_i へのプログラム転送と、シミュレーションクロック毎のシミュレーション開始命令
- ⑪ NP_i で果めに統計量の転送とシミュレーションクロック毎の処理終了報告
- ⑫ バス占有権の要求 (CM アクセスの要求)
- ⑬ バス占有権の許可 (ACK)
- ⑭ CM Read
- ⑮ CM Write

3.2 競合調停方式

NP 間の通信は全て CM 上で行うことは先に述べにが、CM への同時アクセスにより生ずる競合は次に示す様にハードウェア的に調停を行っている。(図2参照)

- ① NP_i ではメモリアクセス要求が生ずると、そのアドレスが CM のアドレス領域かと判断する。
- ② その結果、CM へのアクセスであれば BC に対して $\overline{REQ_i}$ 信号を Low にする。
- ③ BC は CPU クロックの立ち上りの時に $\overline{REQ_i}$ 信号を検出する。

- ④ その時、他の \overline{REQ}_j 信号も low であるかも知れないが、プライオリティエンコードにより $i < j$ ならば i のみが処理され NP_i に \overline{ACK}_i を送り、バス占有権を NP_i に渡す。
- ⑤ 同時に ϕ を Low にしたまま止め、他の NP_s の CPU を止める。
- ⑥ \overline{ACK}_i 信号立上りの時、他 NP_j からの \overline{REQ}_j 信号を検出すると、 \overline{ACK}_j 信号を NP_j に送りバス占有権を NP_j に渡す。従って ACK 信号は、NP がメモリアクセスを終了するだけの十分な長さの時間 low になっている。
- ⑦ 以上のプロセスを行い、 ϕ を high にしている間に ACK 信号の立上り時に REQ 信号を検出しなくなると ϕ を low にすると NP_s の動作を開始する。

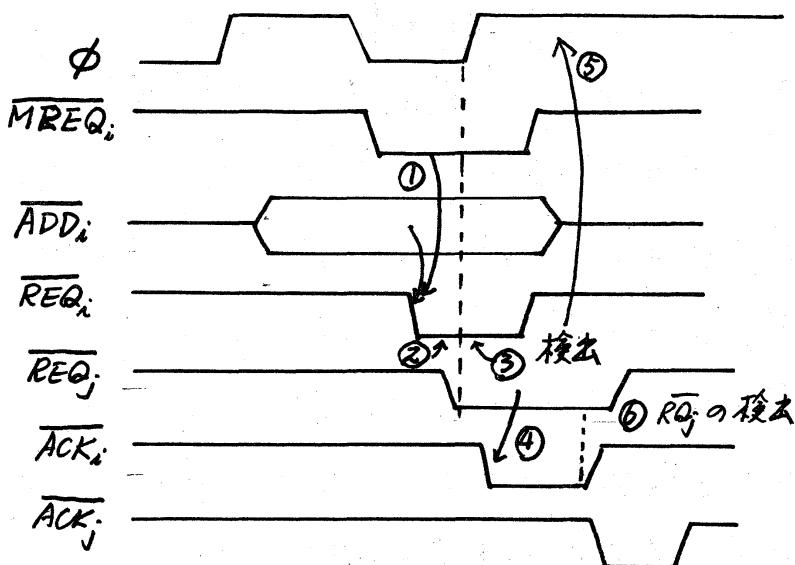


図2 BC タイミングチャート

4. ソフトウェア構成

各プロセッサのアドレス空間は次の様に割り当てている。

CM	8000 ~ FFFF	R/W	32K byte
NP	7000 ~ 7FFF	W	4K byte
RNG	6000, 6001	R	2 byte
RAM	4000 ~ 5FFF	R/W	8K byte
SR	6002, 6003	W	2 byte
Monitor	0000 ~ 0FFF	R	4K byte

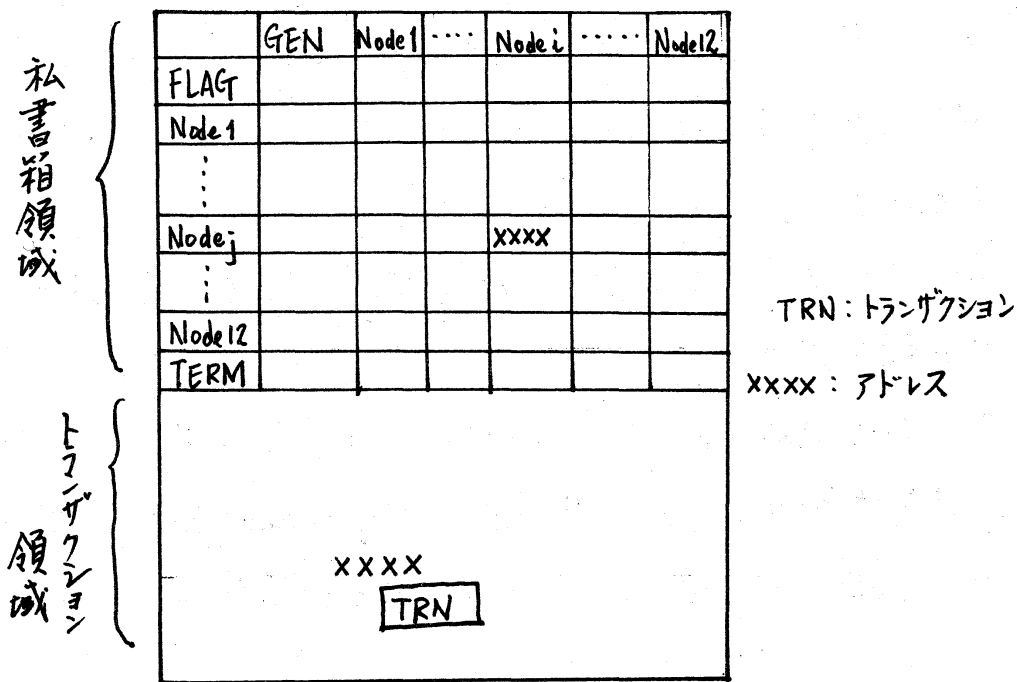
RNG (Random Number Generator) とは、乱数発生器から乱数が入けられる領域。SR は、各 NP にデータを転送するときの NP のセレクトやすべての NP のリセットに用いられる。

4.1 共有メモリ (CM) 管理方式

共有メモリは、図3で示すように各 NP ごとの私書箱領域 (テーブル構成) と CP より generate されたトランザクションを収納するトランザクション領域とに二分される。

各私書箱には、やりとりされるトランザクションが存在するアドレスが書き込まれる。よって各 NP 間でのやりとりは実際のトランザクションではなく、そのトランザクションのあるアドレスである。こうすることにより、NP はトランザクションそのものを読込むのに対し、そのアドレスだけを読込めばよく、メモリアクセス時間が節約できる。

系外に出るトランザクションのアドレスは、図3のTermの項に書き込まれ、CPに知らされる。CPは新しく generate し系内に入るトランザクションとその空きアドレスのところに書き込むと同時に、入力する NP の私書箱の Gen の項にそのアドレスを書く。



GEN: 系外からネットワーク内に入ったトランザクション

TERM: ネットワークから系外へ出ていったトランザクション

FLAG: 該当ノードの待ち行列 (バッファ) がいっぱいかどうかを示す

図3 CMの構成

4.2 通信方式

(i) $NP \leftrightarrow NP$

シミュレーション実行時における NP 間トランザクションの移動は、次の順序で行われる。

- ① CP からシミュレーション時刻での開始命令がくると NP は CM 内の自分の私書箱の内容をサーチし、自分宛のトランザクションがあればそれらのアドレスと全て、各 NP にあるローカルメモリに取込む (Queue に加える) (Post read)
- ② すべての NP の Post read が完了すると、各 NP は並列に処理を行う。
- ③ 処理過程において、他 NP へのトランザクション転送の必要性が生じると、相手ノードの Queue がいっぱいかどうかを調べて、空いていると、相手私書箱内にそのアドレスを記入する (Post write)

(ii) $CP \leftrightarrow NP$

CP から NP への通信は、系外からノードへのトランザクションの入力、 NP から CP への通信は、 NP で集めた統計量の OPU への転送などがあるが、これらも全て CM を通して行われる。

4.3 実行制御

実際シミュレーションを進めていく上で、HASS-QNが必要とするシステムパラメータは

- (1) システム構成(ノード間接続情報)
- (2) トランザクションのノード間遷移確率
- (3) 客の系外から各ノードへ到着するレート
- (4) 処理装置の処理時間分布
- (5) 各ノードでの許容最大待行列
- (6) 統計情報を取り出す間隔

などで、これらをユーザーにUIPより入力してもらい、あとは全ての実行制御はHASS-QNが自動的に行う。

4.4 統計処理

HASS-QNでは、統計処理はすべてOPUが行う。OPUはCPから送られる統計量をフロッピーディスクに格納し、編集出力プログラムがCRT上にリアルタイム(シミュレーション実行時に)に出力する。ユーザーが指定する時点で、取出せる統計情報としては

- (1) 各待行列の長さの分布
- (2) 各待行列の長さの平均
- (3) サーバー利用率
- (4) システム全体のスループット

- (5) トランザクションの平均レスポンスタイム
 - (6) ブロッキング率
 - (7) システムデッドロックに関する情報
- などである。

5. あとがき

ネットワーク形待ち行列にモデル化できるシステムに対し効率の良いシミュレーションが行える HASS-QN の開発について述べてきた。

現在のところ、HASS-QN が対象とし得るモデルは比較的に限られたものとなっているが、今後各ノードが複数窓口を有する場合、処理時間分布としてユーザーからヒストグラムの的に与えにようなものに対しても扱えるよう機能向上を図る予定である。

また、ユーザーが NP, CP のプログラムを容易に開発できるようにすることも同時に考えている。そうすることにより単に QN の解析に役立つというだけでなく、例えば分散型データベースの同期アルゴリズムの評価、システムデッドロック回避アルゴリズムの評価、コンピュータネットワークにおける高価プロトコルの検証などに利用できるようになると考えている。

謝辞 HASS-QNの製作に対し多大の協力を得た高島研究室の諸氏に感謝いたします。

(参考文献)

1. 橋田温, 川島幸之助 "待行列ネットワークモデルによる計算機システムの性能評価" 情報処理, 21, 7, 1980
2. 高橋豊, 宮原秀夫, 長谷川利治 "待行列網理論" システムと制御, 22, 12, 1978
3. J. K. Peacock, et al "Distributed Simulation Using a Network of Processors" Computer Networks, 3, 1, 1979
4. 稲岸久由他 "複合マイコンプロセッサによる並列処理形通信網シミュレータ" 信学技報, EC-79-78, 1980
5. 竹之内博夫他 "複合マイコンによる並列処理シミュレータ" マイコンコンピュータ応用国際コンファレンス'80, 1980
6. 中川徹他 "待行列システム・シミュレーションにおける並列処理" 情報処理学会, 計算機アーキテクチャ研究会資料, 1979
7. 小林信裕他 "分散型シミュレータ KDSS-I の設計と試作" 情報処理学会第21回全国大会資料
8. 西田竹志他 "待行列網シミュレータ" 信学技報 EC80-52, 1980